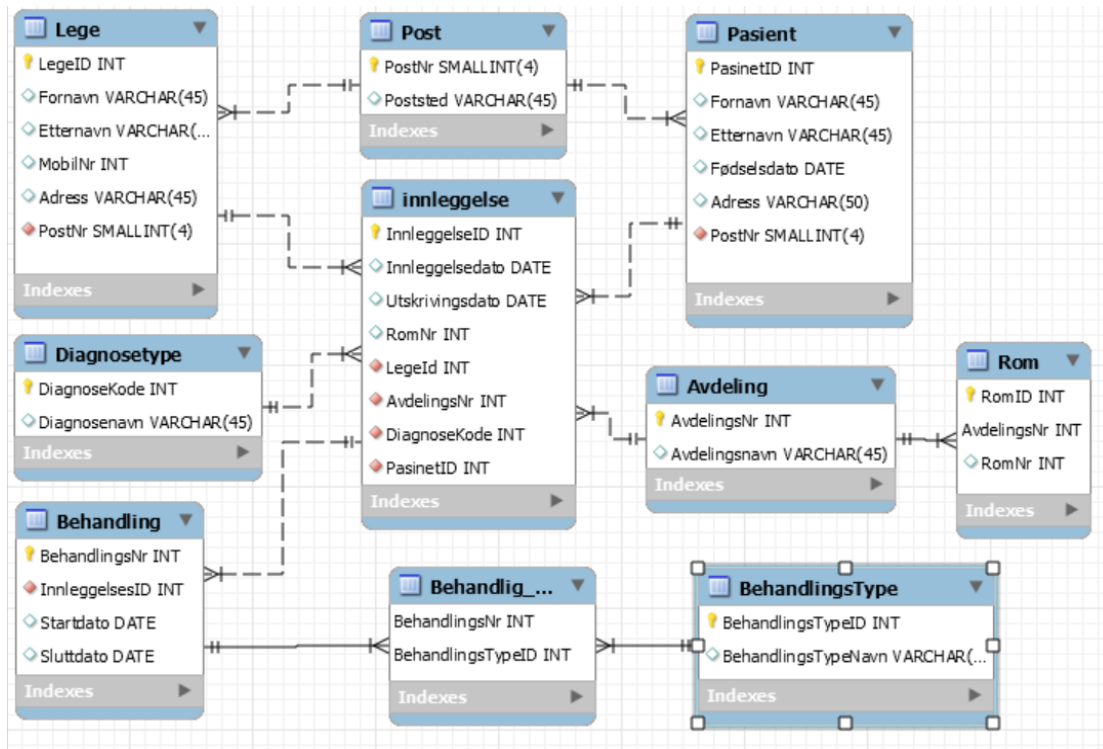


Oblig 3 - Opprette database

Almuthanna Jamal Aljajan

Data modell:



Beskrivelse:

Først fikset jeg noen feil i ER-diagram fra forrige oblig som du har anbefalte å rette. Deretter opprettet jeg selve databasen ved å bruke «Forward Engineer» verktøy som finnes i MYSQL workbench's . Dette verktøyet operetter et SQL-script for opprettelse av database basert på entiteter, attributter og relasjoner som finnes i ER-diagrammet. Jeg har gjort noen endringer i skripten til Forward Engineer ved å bruke AUTO_INCREMENT for å genere Id-en til forskjellige tabeller for å unngå å gjøre det manuelt for hver eneste rad. For å fylle tabeller med data, startet jeg med fylling av tabeller som inneholder ikke fremmedenøkler og som ikke er avhengig av andre tabeller og avsluttet jeg med fylling av koblingstabeller siden de er avhengige av andre tabeller.

Forward Engineer:

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

--
```

```

-- Schema oblig3_Almuthanna
-- -----

-- Schema oblig3_Almuthanna
-- -----
CREATE SCHEMA IF NOT EXISTS `oblig3_Almuthanna` DEFAULT CHARACTER SET utf8
;
USE `oblig3_Almuthanna` ;

-- -----
-- Table `oblig3_Almuthanna`.`Post`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Post` (
  `PostNr` SMALLINT(4) NOT NULL,
  `Poststed` VARCHAR(45) NULL,
  PRIMARY KEY (`PostNr`))
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Lege`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Lege` (
  `LegeID` INT AUTO_INCREMENT,
  `Fornavn` VARCHAR(45) NULL,
  `Etternavn` VARCHAR(45) NULL,
  `MobilNr` INT NULL,
  `Adress` VARCHAR(45) NULL,
  `PostNr` SMALLINT(4) NOT NULL,
  PRIMARY KEY (`LegeID`),
  INDEX `fk_Lege_Post1_idx` (`PostNr` ASC) VISIBLE,
  CONSTRAINT `fk_Lege_Post1`
    FOREIGN KEY (`PostNr`)
      REFERENCES `oblig3_Almuthanna`.`Post` (`PostNr`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Avdeling`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Avdeling` (
  `AvdelingsNr` INT AUTO_INCREMENT,
  `Avdelingsnavn` VARCHAR(45) NULL,
  PRIMARY KEY (`AvdelingsNr`))
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Diagnosetype`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Diagnosetype` (
  `DiagnoseKode` INT AUTO_INCREMENT,
  `Diagnosenavn` VARCHAR(45) NULL,
  PRIMARY KEY (`DiagnoseKode`))
ENGINE = InnoDB;

```

```

-- Table `oblig3_Almuthanna`.`Pasient`
-----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Pasient` (
  `PasinetID` INT AUTO_INCREMENT,
  `Fornavn` VARCHAR(45) NULL,
  `Etternavn` VARCHAR(45) NULL,
  `Fødselsdato` DATE NULL,
  `Adress` VARCHAR(50) NULL,
  `PostNr` SMALLINT(4) NOT NULL,
  PRIMARY KEY (`PasinetID`),
  INDEX `fk_Pasient_Post1_idx` (`PostNr` ASC) VISIBLE,
  CONSTRAINT `fk_Pasient_Post1`
    FOREIGN KEY (`PostNr`)
      REFERENCES `oblig3_Almuthanna`.`Post` (`PostNr`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`innleggelse`
-----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`innleggelse` (
  `InnleggelseID` INT AUTO_INCREMENT,
  `Innleggesdato` DATE NULL,
  `Utskrivingsdato` DATE NULL,
  `RomNr` INT NULL,
  `LegeId` INT NOT NULL,
  `AvdelingsNr` INT NOT NULL,
  `DiagnoseKode` INT NOT NULL,
  `PasinetID` INT NOT NULL,
  PRIMARY KEY (`InnleggelseID`),
  INDEX `fk_innleggelse_Lege_idx` (`LegeId` ASC) VISIBLE,
  INDEX `fk_innleggelse_Avdelin1_idx` (`AvdelingsNr` ASC) VISIBLE,
  INDEX `fk_innleggelse_Diagnosetype1_idx` (`DiagnoseKode` ASC) VISIBLE,
  INDEX `fk_innleggelse_Pasinet1_idx` (`PasinetID` ASC) VISIBLE,
  CONSTRAINT `fk_innleggelse_Lege`
    FOREIGN KEY (`LegeId`)
      REFERENCES `oblig3_Almuthanna`.`Lege` (`LegeID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_innleggelse_Avdelin1`
    FOREIGN KEY (`AvdelingsNr`)
      REFERENCES `oblig3_Almuthanna`.`Avdeling` (`AvdelingsNr`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_innleggelse_Diagnosetype1`
    FOREIGN KEY (`DiagnoseKode`)
      REFERENCES `oblig3_Almuthanna`.`Diagnosetype` (`DiagnoseKode`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_innleggelse_Pasinet1`
    FOREIGN KEY (`PasinetID`)
      REFERENCES `oblig3_Almuthanna`.`Pasient` (`PasinetID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Behandling`

```

```

-----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Behandling` (
  `BehandlingsNr` INT AUTO_INCREMENT,
  `InnleggesID` INT NOT NULL,
  `Startdato` DATE NULL,
  `Sluttdato` DATE NULL,
  PRIMARY KEY (`BehandlingsNr`),
  CONSTRAINT `fk_Behandling_innlegges1`
    FOREIGN KEY (`InnleggesID`)
    REFERENCES `oblig3_Almuthanna`.`innlegges` (`InnleggesID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`BehandlingsType`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`BehandlingsType` (
  `BehandlingsTypeID` INT AUTO_INCREMENT,
  `BehandlingsTypeNavn` VARCHAR(45) NULL,
  PRIMARY KEY (`BehandlingsTypeID`))
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Behandlig_has_behandlingstype`
-- -----
CREATE TABLE IF NOT EXISTS
`oblig3_Almuthanna`.`Behandlig_has_behandlingstype` (
  `BehandlingsNr` INT NOT NULL,
  `BehandlingsTypeID` INT NOT NULL,
  PRIMARY KEY (`BehandlingsNr`, `BehandlingsTypeID`),
  INDEX `fk_DelBehandling_BehandlingsType1_idx` (`BehandlingsTypeID` ASC)
  VISIBLE,
  CONSTRAINT `fk_DelBehandling_Behandling1`
    FOREIGN KEY (`BehandlingsNr`)
    REFERENCES `oblig3_Almuthanna`.`Behandling` (`BehandlingsNr`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_DelBehandling_BehandlingsType1`
    FOREIGN KEY (`BehandlingsTypeID`)
    REFERENCES `oblig3_Almuthanna`.`BehandlingsType` (`BehandlingsTypeID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig3_Almuthanna`.`Rom`
-- -----
CREATE TABLE IF NOT EXISTS `oblig3_Almuthanna`.`Rom` (
  `RomID` INT AUTO_INCREMENT,
  `AvdelingsNr` INT NOT NULL,
  `RomNr` INT NULL,
  PRIMARY KEY (`RomID`, `AvdelingsNr`),
  INDEX `fk_Rom_Avdeling1_idx` (`AvdelingsNr` ASC) VISIBLE,
  CONSTRAINT `fk_Rom_Avdeling1`
    FOREIGN KEY (`AvdelingsNr`)
    REFERENCES `oblig3_Almuthanna`.`Avdeling` (`AvdelingsNr`)
    ON DELETE NO ACTION

```

```

        ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

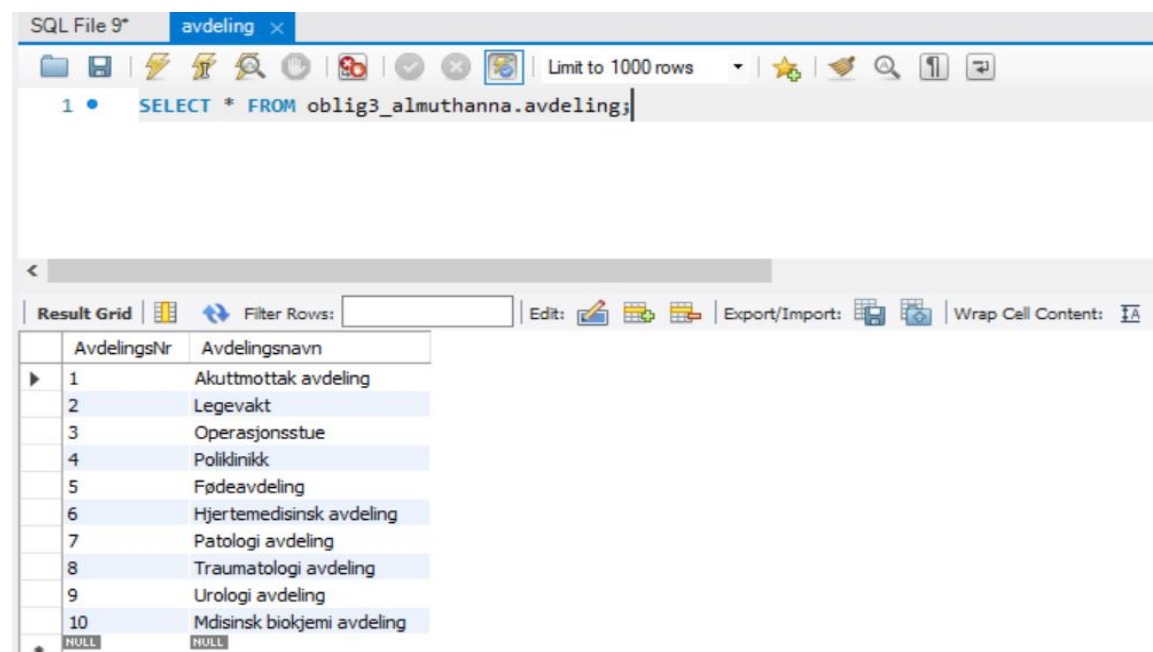
Kode for å fylle tabeller:

Avdeling tabell:

```

Insert into avdeling (Avdelingsnavn) values
('Akuttmottak avdeling '),
('Legevakt'),
('Operasjonsstue'),
('Poliklinikk'),
('Fødeavdeling'),
('Hjertemedisinsk avdeling'),
('Patologi avdeling'),
('Traumatologi avdeling'),
('Urologi avdeling'),
('Mdisinsk biokjemi avdeling');

```



SQL File 9* avdeling x

Limit to 1000 rows

1 • SELECT * FROM oblig3_almuthanna.avdeling;

Result Grid

	AvdelingsNr	Avdelingsnavn
▶	1	Akuttmottak avdeling
	2	Legevakt
	3	Operasjonsstue
	4	Poliklinikk
	5	Fødeavdeling
	6	Hjertemedisinsk avdeling
	7	Patologi avdeling
	8	Traumatologi avdeling
	9	Urologi avdeling
	10	Mdisinsk biokjemi avdeling
	NULL	NULL

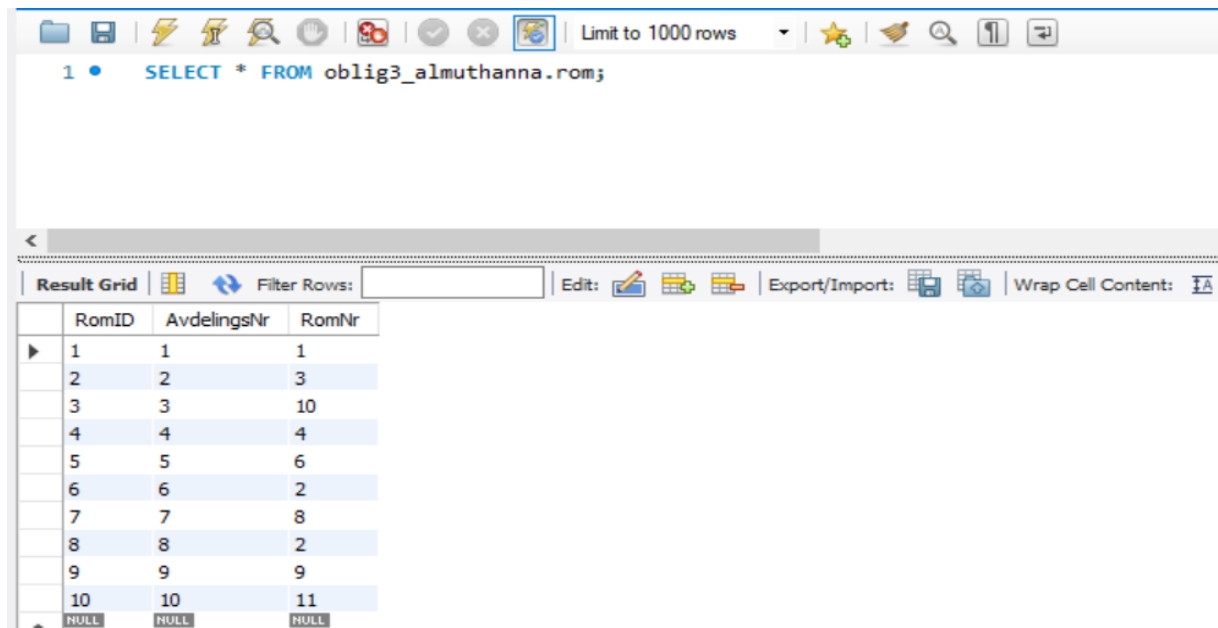
Rom Tabell:

```

insert into rom (AvdelingsNr, RomNr) values
(1, 1),
(2, 3),
(3, 10),

```

```
(4, 4),
(5, 6),
(6, 2),
(7, 8),
(8, 2),
(9, 9),
(10, 11);
```



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text area: `1 • SELECT * FROM oblig3_almuthanna.rom;`. Below the query area, there is a horizontal scrollbar. At the bottom, there is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The 'Result Grid' icon is selected, and the results are displayed in a table below.

	RomID	AvdelingsNr	RomNr
▶	1	1	1
	2	2	3
	3	3	10
	4	4	4
	5	5	6
	6	6	2
	7	7	8
	8	8	2
	9	9	9
	10	10	11
	NULL	NULL	NULL

Post tabell:

```
insert into post (PostNr, Poststed) values
( 1400, 'Ski' ),
( 1430, 'Ås' ),
( 1429, 'Vinterbro' ),
( 1440, 'Drøbak' ),
( 1500, 'Moss' ),
( 1482, 'Nittedal' ),
( 1474, 'Lørenskog' ),
( 1454, 'Fagerstrand' ),
( 1456, 'Nesoddtangen' ),
( 1600, 'Fredrikstad' );
```

1 • **SELECT * FROM oblig3_almuthanna.post;**

PostNr	Poststed
1400	Ski
1429	Vinterbro
1430	Ås
1440	Drøbak
1454	Fagerstrand
1456	Nesoddtangen
1474	Lørenskog
1482	Nittedal
1500	Moss
1600	Fredrikstad
NULL	NULL

Lege tabell:

```
alter table lege auto_increment=100;
insert into Lege (Fornavn,Etternavn,MobilNr,Adress,PostNr) values
('Martin', 'Langhås', 93345677, 'Ødegårdsveien 7', 1400),
('Catrine', 'William', 98877234, 'Vilbergveien 5', 1430 ),
('Ole', 'Hansen', 93453429, 'elgveien 1',1429),
('Kristoffer', 'Nilsen', 45783444, 'Laabyveien 22',1440),
('Arthur', 'Andersen', 86512453, 'Karl Johans gate 5',1500),
('Husam', 'Khudhair', 53289765, 'Oskleiva 16',1482),
('Jonas', 'Eliassen', '93318772', 'Herfordts gate 2',1474),
('Erik', 'Larsen', 45671200, 'Axel Heibergs gate 2',1454),
('Daria', 'Demidova', 43348998, 'Stein Jordet, Ridehusgata 12',1456),
('Björn', 'Andersson', 54897634, 'Knipleveien 8C', 1600);
```

SQL File 9* avdeling rom post lege

Limit to 1000 rows

1 • SELECT * FROM oblig3_almuthanna.lege;

Result Grid Filter Rows: Edit: Export/Import:

	LegeID	Fornavn	Etternavn	MobilNr	Adress	PostNr
▶	100	Martin	Langhås	93345677	Ødegårdsveien 7	1400
	101	Catrine	William	98877234	Vilbergveien 5	1430
	102	Ole	Hansen	93453429	elgveien 1	1429
	103	Kristoffer	Nilsen	45783444	Laabyveien 22	1440
	104	Arthur	Andersen	86512453	Karl Johans gate 5	1500
	105	Husam	Khudhair	53289765	Oskleiva 16	1482
	106	Jonas	Eliassen	93318772	Herfordts gate 2	1474
	107	Erik	Larsen	45671200	Axel Heibergs gate 2	1454
	108	Daria	Demidova	43348998	Stein Jordet, Ridehusgata 12	1456
	109	Björn	Andersson	54897634	Knipleveien 8C	1600
★	NULL	NULL	NULL	NULL	NULL	NULL

Pasient tabell:

```
alter table pasient auto_increment=200;
insert into pasient (Fornavn,Etternavn,Fødselsdato,Adress,PostNr) values
('Haakon','Stangborli','2003-04-05','Henrik Ibsens vei 2',1482),
('Charis','Barneveld Schille','1994-12-03','Solveien 23',1600),
('Benjamin','Daniel','2014-10-08','Prestelandet 16A',1440),
('Andreas','Thauland','1995-07-13','Ridehusgata 12',1430),
('Oskar','Lynum','1999-06-20','Enggata 97B',1474),
('Dennis','Olsen','1957-02-28','Litles gate 10',1400),
('Simen','Oskarsen','2012-08-13','Cicignongata 20',1500),
('Emilie','Nilsen','2010-12-23','Myreveien 20',1430),
('Simon','Myhre','1986-03-26','Bydalen alle 13',1429),
('Magnus','Munthe','1945-07-19','Roald Amundsens gate 54',1430);
```


1 • **SELECT * FROM oblig3_almuthanna.pasient;**

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	PasinetID	Fornavn	Etternavn	Fødselsdato	Adress	PostNr
▶	200	Haakon	Stangborli	2003-04-05	Henrik Ibsens vei 2	1482
	201	Charis	Barneveld Schille	1994-12-03	Solveien 23	1600
	202	Benjamin	Daniel	2014-10-08	Prestelandet 16A	1440
	203	Andreas	Thauland	1995-07-13	Ridehusgata 12	1430
	204	Oskar	Lynum	1999-06-20	Enggata 97B	1474
	205	Dennis	Olsen	1957-02-28	Litles gate 10	1400
	206	Simen	Oskarsen	2012-08-13	Cicignongata 20	1500
	207	Emilie	Nilsen	2010-12-23	Myreveien 20	1430
	208	Simon	Myhre	1986-03-26	Bydalen alle 13	1429
	209	Magnus	Munthe	1945-07-19	Roald Amundsens gate 54	1430
*	NULL	NULL	NULL	NULL	NULL	NULL

Diagnosetype tabell:

```
alter table diagnosetype auto_increment=300;
insert into diagnosetype (Diagnosenavn) values
('Bipolar'),
('Depresjon'),
('ADHD'),
('Hypokondri'),
('Migrene'),
('Parkinsonisme'),
('Psoriasis'),
('Diabetes mellitus'),
('kreft'),
('Slitasjegikt'),
('Hjertesvikt'),
('Epilepsi'),
('Kronisk obstruktiv'),
('Hørselsvekkelse'),
('Angina pectoris'),
('Schizofreni'),
('Personlighetsforstyrrelse');
```

1 • SELECT * FROM oblig3_almuthanna.diagnosetype;

DiagnoseKode	Diagnosenavn
300	Bipolar
301	Depresjon
302	ADHD
303	Hypokondri
304	Migrene
305	Parkinsonisme
306	Psoriasis
307	Diabetes mellitus
308	kreft
309	Slitasjegikt
310	Hjertesvikt
311	Epilepsi
312	Kronisk obstruktiv
313	Hørselsvekkelse
314	Angina pectoris
315	Schizofreni
316	Personlighetsfo...

Innleggelse tabell

```
alter table innleggelse auto_increment=400;
insert into innleggelse
(Innleggelsesdato,Utskrivingsdato,RomNr,LegeId,AvdelingsNr,DiagnoseKode,Pasi
netID) values
('2021-01-20','2021-02-03',11,100,1,300,200),
('2020-12-11','2021-01-25',22,101,2,301,201),
('2021-01-09','2021-01-23',33,102,3,302,202),
('2021-01-04','2021-01-30',44,103,4,303,203),
('2021-02-01','2021-02-19',55,104,5,304,204),
('2021-02-04','2021-02-11',66,105,6,305,205),
('2021-02-10','2021-03-29',77,106,7,306,206),
('2021-02-20','2021-03-17',88,107,8,307,207),
('2021-02-28','2021-02-16',99,108,9,308,208),
('2021-02-03','2021-02-21',1010,109,10,309,209);
```

1 • `SELECT * FROM oblig3_almuthanna.innleggelse;`

Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	InnleggelseID	Innleggelsesdato	Utskrivingsdato	RomNr	LegeId	AvdelingsNr	DiagnoseKode	PasinetID
▶	400	2021-01-20	2021-02-03	11	100	1	300	200
	401	2020-12-11	2021-01-25	22	101	2	301	201
	402	2021-01-09	2021-01-23	33	102	3	302	202
	403	2021-01-04	2021-01-30	44	103	4	303	203
	404	2021-02-01	2021-02-19	55	104	5	304	204
	405	2021-02-04	2021-02-11	66	105	6	305	205
	406	2021-02-10	2021-03-29	77	106	7	306	206
	407	2021-02-20	2021-03-17	88	107	8	307	207
	408	2021-02-28	2021-02-16	99	108	9	308	208
	409	2021-02-03	2021-02-21	1010	109	10	309	209
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Behandling tabell:

```
alter table behandling auto_increment=500;
insert into behandling (InnleggelsesID, Startdato, Sluttdato) values
(400, '2021-01-23', '2021-01-29'),
(401, '2021-01-01', '2021-01-03'),
(402, '2021-01-12', '2021-01-20'),
(403, '2021-01-04', '2021-01-28'),
(404, '2021-02-03', '2021-02-17'),
(405, '2021-02-11', '2021-02-23'),
(406, '2021-02-12', '2021-03-22'),
(407, '2021-02-25', '2021-03-11'),
(408, '2021-03-1', '2021-02-02'),
(409, '2021-02-09', '2021-02-10');
```

1 • **SELECT * FROM oblig3_almuthanna.behandling;**

Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	BehandlingsNr	InnleggesID	Startdato	Sluttdato
▶	500	400	2021-01-23	2021-01-29
	501	401	2021-01-01	2021-01-03
	502	402	2021-01-12	2021-01-20
	503	403	2021-01-04	2021-01-28
	504	404	2021-02-03	2021-02-17
	505	405	2021-02-11	2021-02-23
	506	406	2021-02-12	2021-03-22
	507	407	2021-02-25	2021-03-11
	508	408	2021-03-01	2021-02-02
	509	409	2021-02-09	2021-02-10
*	NULL	NULL	NULL	NULL

Behandlingstype tabell:

```
alter table behandlingstype auto_increment=600;
Insert into behandlingstype (BehandlingsTypeNavn) values
('kirurgi'),
('Hypnoterapi'),
('Kiropraktor'),
('Fysioterapi'),
('Massasje'),
('Trykkbølgebehandling'),
('Manuellterapi'),
('Medisinsk'),
('injeksjon'),
('Elektroterapi');
```

SQL File 9* | avdeling | rom | post | lege | pasient | diagnosetype | diagnosetype | in

1 • **SELECT * FROM oblig3_almuthanna.behandlingstype;**

Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	BehandlingsTypeID	BehandlingsTypeNavn
▶	600	kirurgi
	601	Hypnoterapi
	602	Kiropraktor
	603	Fysioterapi
	604	Massasje
	605	Trykkbølgebehandling
	606	Manuellterapi
	607	Medisinsk
	608	injeksjon
	609	Elektroterapi
*	NULL	NULL

behandlig_has_behandlingstype tabell:

```
Insert into behandlig_has_behandlingstype (BehandlingsNr,BehandlingsTypeID)
values
(500,600),
(501,601),
(502,602),
(503,603),
(504,604),
(505,605),
(506,606),
(507,607),
(508,608),
(509,609);
```

The screenshot shows a database management tool interface. At the top, there are several tabs: 'le 9*', 'avdeling', 'rom', 'post', 'lege', 'pasient', 'diagnosetype', 'diagnosetype', and 'innleggels'. Below the tabs is a toolbar with various icons. The main area displays a SQL query: `1 • SELECT * FROM oblig3_almuthanna.behandlig_has_behandlingstype;`. Below the query, there is a 'Result Grid' section. It includes a 'Filter Rows:' input field and a toolbar with icons for 'Edit', 'Export/Import', and 'Wrap Cell Content:'. The result grid shows a table with two columns: 'BehandlingsNr' and 'BehandlingsTypeID'. The data rows are numbered 1 to 10, with the last row showing 'NULL' for both columns.

	BehandlingsNr	BehandlingsTypeID
▶	500	600
	501	601
	502	602
	503	603
	504	604
	505	605
	506	606
	507	607
	508	608
	509	609
•	NULL	NULL